arXiv:2611.30170v1 [cs.CV] 23 Nov 2026

GUI-Q：面向图形用户界面智能体的任务敏感粒度与关系保持量化

Tianqi Li[1,*], Yang Wang[2,*], Xin He[3], LeZhang[4], YunLiu[1,5,6,†]

VCIP，南开大学计算机学院[2]北京化工大学数理学院[3]天津理工大学计算机科学与工程学院[4]电子科技大学信息与通信工程学院[5]南开大学先进交叉学科研究院[6]南开国际先进研究院，深圳福田[*]共同贡献。[†]通讯作者。

摘要

图形用户界面（*GUI*）智能体能够自动与移动端、网页和桌面界面交互。然而，它们庞大的内存占用和反复的模型调用阻碍了其在资源受限设备上的部署，因此需要在无需完整重新训练的情况下进行训练后量化（*PTQ*）。量化可能降低 *GUI* 响应准确率，并扭曲 *GUI* 目标选择所需的指令-视觉亲和度和局部空间关系。微小的坐标误差可能导致点击落空，而早期动作错误可能改变后续界面状态并扰乱任务执行。仅凭局部重建误差无法直接评估这些响应和关系是否得到保留。我们提出***GUI-Q***，一个采用 *4* 位权重和 *16* 位激活（*W4A16*），旨在保持响应和关系的训练后量化框架。任务敏感粒度分配（***TSGA***）通过结合响应敏感度与激活加权失真的实测降幅，在尺度字节预算下分配更细的分组。随后，关系保持对称性选择（***RPSS***）选择对称或非对称网格，以保留图形用户界面中的指令-视觉亲和度和局部空间关系。在三个支持图形用户界面的模型上，与评测中表现最强的 *4* 位配置相比，*GUI-Q* 将平均定位目标命中率提高了 *8.04–12.55* 个百分点，同时相较于脑浮点数（*BF16*），将静态峰值图形处理器（*GPU*）内存减少了 *49.9–56.2%*。采用打包 *4* 位执行的固定输出回放可带来 *1.7–2.8*×倍模型调用加速。代码即将发布。

1. 引言

图形用户界面代理将指令和屏幕截图转化为移动端、网页和桌面界面上的动作。除了理解图像和语言 [8, 24, 25], 它们还必须定位目-能够准确获取信息并正确指定动作参数。坐标出现微小误差就可能导致点击失败，而早期的错误可能会改变后续步骤中的界面状态。

近期的图形用户界面智能体 [21, 26, 35] 依赖大型视觉语言模型和重复的模型调用，因此产生了巨大的内存和推理成本。对于 7–8B 模型，FP32 权重需要约 26–30 吉比字节的内存。即使采用脑浮点数（BF16），在我们的静态评估中，峰值 GPU 内存分配也达到 15.75–16.67 吉比字节（第4.6节）。这些资源需求使得在内存和计算资源有限的智能手机及其他边缘设备上进行本地部署更加困难。训练后量化（PTQ）提供了一种实用方案，可在复用预训练模型而无需完全重新训练的同时，减少权重存储。我们着重研究4位权重量化（W4），以降低模型成本，同时保持准确的目标选择和动作生成。这促使我们考察量化如何影响图形用户界面响应、指令与目标之间的关联，以及界面中相邻图形用户界面元素之间的空间关系。

现有训练后量化方法采用重建和基于激活的缩放 [12, 19]。近期方法引入了端损失引导、模态或令牌敏感度，以及动作保真度[1, 13, 16, 27, 30]。这些进展提供了有用的任务感知信号，但识别敏感模块并不能确定在哪些位置采用更细粒度分组最有益。对于图形用户界面智能体而言，仅采用序列平均目标并不能明确平衡数值内容与响应的其余部分；而局部重建误差也无法直接衡量指令与目标之间的关系以及空间关系是否得到保持。图 1 所示的落地和动作差距，促使我们在图形用户界面量化中关注响应保留和关系保持。

图形用户界面量化应在有限的尺度字节预算下保留响应，并保留指令-视觉亲和度

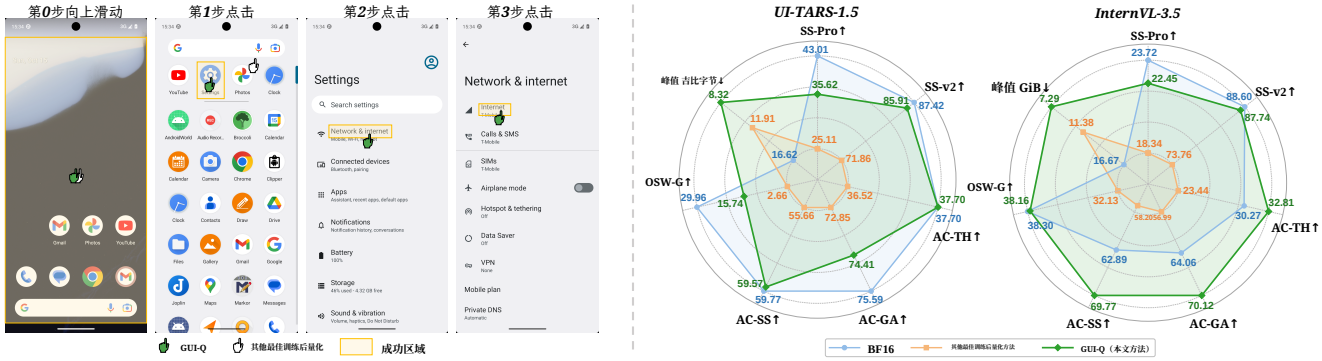


图1. 任务与跨模型比较。左：四步轨迹对比。右：BF16、最强的训练后量化基线方法，以及 GUI-Q 在六项准确率指标和峰值内存方面的比较。

以及选择图形用户界面目标所需的局部空间关系。这些目标指导我们对粒度和对称性的选择。粒度决定哪些权重共用一个缩放因子。每组包含64个权重（G64）时，范围调整比每组128个权重（G128）更局部，但尺度元数据会增加一倍。因此，应将更细的分组分配到最有利于保留图形用户界面响应的位置。对称性决定每组内部的网格：非对称映射可以适应偏移范围，但代价是需要零点元数据。在这些网格之间进行选择，需要评估它们对图形用户界面中的指令-视觉亲和度和局部空间关系有何影响。补充表S85（第S13节）比较了这两种量化方案在所评估图形用户界面任务上的准确率。

我们提出了**GUI-Q**，这是一种采用 4 位权重和 16位激活值（W4A16）的框架。任务敏感粒度分配（**TSGA**）根据完整响应、含数字词元和其余词元估算模块敏感度。它将这一敏感度与从 G128 降至 G64 所带来的激活加权失真实测降幅相结合，在尺度字节预算约束下，按模块分配更细的分组。此类无标签校准信号利用冻结模型自身的响应。在由此确定的分组大小下，关系保持对称性选择（**RPSS**）依据对指令-视觉亲和度及图形用户界面局部空间关系的保持程度，选择对称或非对称网格。

在 UI-TARS-1.5-7B、OS-Atlas-Base-7B 和 InternVL3.5-8B 上，与评测中最强的 W4配置相比，GUI-Q 的平均落地 TargetHit（TH）分别提高了 12.55、9.62 和 8.04 分；与 BF16 相比，静态峰值 GPU 内存减少了 49.9–56.2%。另在 AndroidWorld 上对 AgentCPM-GUI 进行的独立评测中，[22] 严格成功率提高了 7.2 分。打包的 4 位模型调用在我们的固定输出重放中运行速度提升了 1.7–2。8×倍。

我们的工作主要有以下三项贡献：

• 一种面向图形用户界面智能体的 W4A16 训练后量化框架，在尺度字节预算和非对称模块的上限约束下，结合粒度分配与对称性选择。

• TSGA 根据响应敏感度和失真降低分配更细的分组，以保留图形用户界面响应同时共享用于存储量化尺度的预算。

• RPSS 选择对称或非对称网格，以保留选择正确图形用户界面目标所需的指令-视觉亲和度和局部空间关系。

2. 相关工作

图形用户界面理解与智能体。 图形用户界面系统已从视觉模板和界面编码器发展到截图定位与通用交互 [2, 3, 5, 31]。如今，网页、移动端和桌面端基准测试同时评估单步定位和多步执行 [9, 14, 15, 18, 28,29, 36]。UI-TARS 和 OS-Atlas 提供了具有代表性的、面向图形用户界面的检查点，用于定位和动作生成；而 InternVL3.5 则提供了具备 GUI交互能力的通用多模态检查点。静态定位评估目标定位，而闭环评估则捕捉动作错误如何影响后续观测结果和任务结果。这些要求促使我们在量化图形用户界面智能体时保留目标选择和动作生成能力。

训练后量化。 训练后量化方法会优化权重舍入 [12, 20], , 利用激活值重新缩放权重 [19], , 或学习校准变换 [23]。相关工作还研究了激活量化中的令牌组约束[17]。基于敏感度的方法使用梯度、曲率、扰动和最终损失信号 [4, 10, 11, 13]。AutoRound 由 SignRound 公式及其实现系列代表 [6]。SignRoundV2 将梯度引导的分配与尺度初始化和调优相结合，这些方法也适用于固定的位宽 [7]。单凭敏感度无法判断某个模块能从更细粒度分组中获益多少，而单凭失真降低也无法衡量其对图形用户界面响应的影响。GUI-Q 将响应敏感度与实测的 G128 至 G64 失真降低相结合，在共享的缩放因子字节预算下分配更细粒度的分组。

多模态与动作感知量化。多模态训练后量化研究层间依赖关系、模态平衡、不对称性补偿、位宽分配和令牌敏感度 [16, 27, 32, 34]。在具身控制领域，QVLA 和

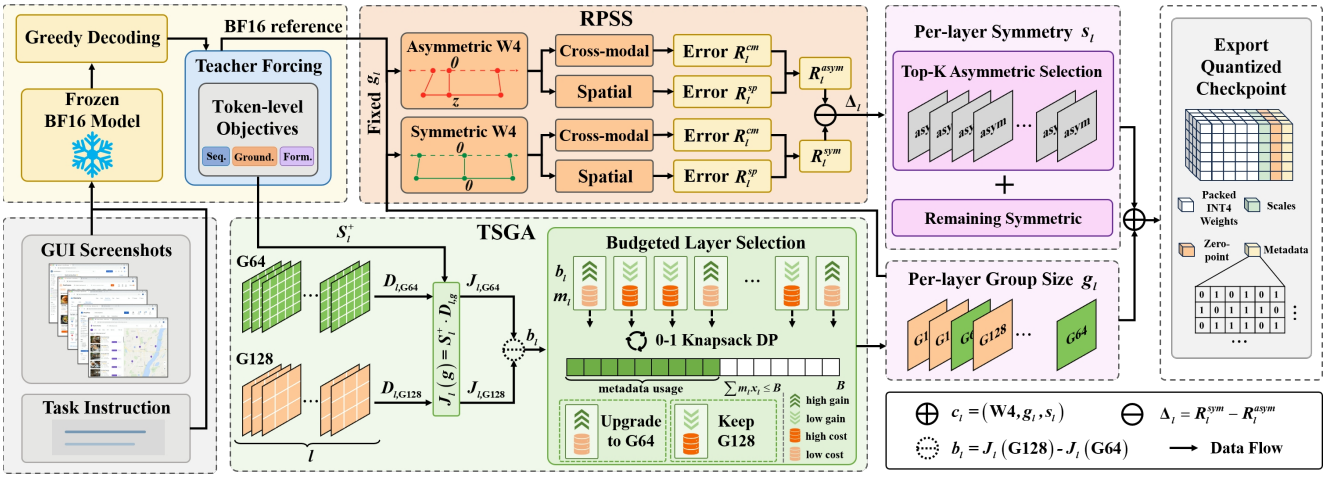


图 2. GUI-Q 概览：TSGA 分配任务敏感的粒度，而 RPSS 选择保留关系的量化器。

ActQuant 使用对动作空间或预测动作的敏感度[1, 30], , 而 QuantVLA 研究面向视觉-语言-动作模型的缩放因子校准训练后量化 [33]。任务敏感度和全精度自生成动作参考也见于以往的训练后量化研究。对于 GUI目标选择，局部重建误差无法直接评估指令-视觉亲和度或局部空间关系的变化。RPSS 通过评估这些关系的保留情况，为 TSGA 分配的分组大小选择对称或非对称网格。

3. 方法

3.1. 问题表述

我们将所有目标线性权重量化为 4位权重，而激活值和非目标组件保持为 BF16。在元数据约束下，我们选择粒度和对称性，以保留用于目标选择的 GUI响应及其关系。

设 f_θ 为冻结的GUI视觉-语言模型，$\mathcal{D}_c = \{(I_n, u_n)\}_{n=1}^{N}$ 为由截图—指令对构成的无标签校准集。对于样本 n 和线性模块 $l \in \mathcal{L}$，$H_{l,n}$ 在所有 T_n 响应位置均有教师强制输入：

$$H_{l,n} \in \mathbb{R}^{T_n \times d_l^{\text{in}}}, \qquad W_l, \widehat{W}_l \in \mathbb{R}^{d_l^{\text{out}} \times d_l^{\text{in}}}, \\ Y_{l,n} = H_{l,n} W_l^{\top} \in \mathbb{R}^{T_n \times d_l^{\text{out}}}. \tag{1}$$

模块 l 使用 $c_l = (4,\ g_l,\ a_l)$，其中 $g_l \in \{64, 128\}$ 和 $a_l \in \{\text{sym},\ \text{asym}\}$ 分别表示分组大小和网格对称性。

粒度与对称性。将一个 G128分组拆分为两个 G64分组，可以分别设置范围；当权重范围不同时，这有助于提高分辨率。对于每个目标权重，G128 和 G64 所需的两个字节缩放因子分别占用 0.125 和 0.25 比特。这些受支持的嵌套选项体现了可测量的权衡，而不预设哪种方案普遍最优。仅凭敏感度无法证明更细粒度分组的益处；仅凭失真降低也无法衡量其对图形用户界面响应的影响。TSGA 结合这两种信号，找出从 G128 切换到 G64 所带来的收益足以抵偿额外缩放因子的模块。

对称网格将范围设为权重绝对值的最大值；非对称网格则增加一个零点，以支持偏移范围。两者都可能改变指令与竞争目标之间的相对亲和度。在指定粒度下，RPSS 会比较这两种网格保留指令-视觉亲和度以及邻近视觉词元之间空间关系的效果。

量化映射。 对于输出行中的每个输入通道分组 w，对称 4位权重使用编码值 $[-7, 7]$、缩放因子$\alpha_s = \max(\max_i |w_i|/7, \epsilon_s)$ 和零点 $z_s = 0$。对于非恒定非对称分组，设 $a = \min(\min_i w_i, 0)$ 和 $b = \max(\max_i w_i, 0)$。其缩放因子和零点为

$$\alpha_a = \max\left(\frac{b-a}{15}, \epsilon_s\right), \\ z_a = \text{clip}\left(\text{rnd}\left(-\frac{a}{\alpha_a}\right), 0, 15\right). \tag{2}$$

给定所选的缩放因子 α、零点 z 和编码范围$q_{\min}, q_{\max}$，每个权重共用的映射为

$$\widehat{w}_i = \alpha\left[\text{clip}(\text{rnd}(w_i/\alpha) + z, q_{\min}, q_{\max}) - z\right]. \tag{3}$$

此处 rnd 表示舍入到最近值，中点取偶。计算编码值和零点之前，尺度会以 BF16 格式存储，其中 $\epsilon_s = 10^{-8}$。在非对称范围中纳入零值可避免单侧分组发生塌缩。补充材料第 S1.3 节介绍了恒定分组、打包和舍入规则。

耦合决策与分阶段选择。TSGA在增量尺度字节预算 B_{scale}下，以共同的对称参考为基准，为所有G128分配分组大小。对于这些分组大小，RPSS从冻结的注意力投影池 K_{asym} 中选择最多 $\mathcal{A}$ 个非对称模块 $\mathcal{C}$。其他模块保持对称。在两项分配均冻结后，共同的舍入后端会校准并导出打包检查点（图 2）。补充材料第S7节分别审计填充、对齐、零点和后端元数据。第 4.3节通过比较独立RPSS得分与联合量化模型在我们评估的下游图形用户界面任务上的观测结果，检验这一分阶段近似方法。

3.2. 任务敏感粒度分配

TSGA对冻结模型响应进行划分，以估计任务加权的可恢复失真。含数字词元仅占校准响应词元的17.15–20.85%。独立的词汇掩码会按照其超出序列平均频率的程度对这些词元进行加权。这些共享词汇掩码近似表示跨输出格式的数值字段及其余响应结构，而无需解析精确坐标或结构化动作。

对于每一对样本，BF16模型以贪心方式生成

$$y_n^* = \mathrm{Greedy}(f_\theta, I_n, u_n). \tag{4}$$

使用原生聊天模板进行教师强制可得到

$$\ell_{n,k} = -\log p_\theta\left(y_{n,k}^* \mid I_n, u_n, y_{n,<k}^*\right). \tag{5}$$

损失计算会排除填充，并在生成时包含序列末尾（EOS）标记。令 $\mathcal{T} = \{\text{seq}、\text{num}、\text{non}\}$以及 $\mathcal{I}_n^{\text{seq}} = \{1, \ldots, T_n\}$。掩码 $\mathcal{B}_n^{\text{num}}$选择解码后的标记形式包含Unicode十进制数字的位置；$\mathcal{B}_n^{\text{non}}$选择其余位置。空词汇掩码会使用响应中的所有标记位置：

$$\mathcal{I}_n^t = \begin{cases} \mathcal{B}_n^t, & |\mathcal{B}_n^t| > 0, \\ \mathcal{I}_n^{\text{seq}}, & |\mathcal{B}_n^t| = 0, \end{cases} \qquad t \in \{\text{num}, \text{non}\}. \tag{6}$$

$$\mathcal{L}_{n,t} = \frac{1}{|\mathcal{I}_n^t|} \sum_{k \in \mathcal{I}_n^t} \ell_{n,k}, \qquad t \in \mathcal{T}. \tag{7}$$

在对所有模型进行测试前，我们固定（p_{seq}、p_{num}、p_{non}）=(0.30、0.50、0.20)。给定$U_{l,t,n} = \partial \mathcal{L}_{n,t} / \partial Y_{l,n}$，TSGA使用以下因子化代理指标估计模块敏感度：

$$\widehat{E}_{l,t} = \frac{1}{N} \sum_{n=1}^{N} \mathrm{mean}(H_{l,n}^{\odot 2}) \, \mathrm{mean}(U_{l,t,n}^{\odot 2}). \tag{8}$$

响应标记损失的梯度会通过视觉前缀和指令前缀传播，但 $H_{l,n}$ 和 $U_{l,t,n}$ 仅保留响应 $\widehat{}$ 行。因此，$E_{l,t}$ 是一个受响应限制的因子化排序代理指标。百分位归一化会在加权前使三个目标得分具有可比性。失真衡量改变网格所能带来的收益。冻结的自生成响应提供一致的校准目标；预言式和全位置控制组用于检验这些近似。按升序排列能量并对并列值取平均排名，可得到百分位数：

$$\widetilde{E}_{l,t} = \frac{\mathrm{rank}_{j \in \mathcal{L}}(\widehat{E}_{j,t}) - 1}{\max(|\mathcal{L}| - 1, 1)}, \tag{9}$$

$$R_l = \sum_{t \in \mathcal{T}} p_t \widetilde{E}_{l,t}, \qquad S_l^+ = R_l + 10^{-4}. \tag{10}$$

在最终舍入校准之前，TSGA 会对称性最小–最大量化进行评分。对于分组大小 g, W_l^{sym}，g 表示量化权重。激活加权失真、敏感度加权目标以及缩放因子存储成本为

$$D_{l,g} = \frac{\sum_n \|H_{l,n}(W_l - \overline{W}_{l,g}^{\text{sym}})^\top\|_F^2}{\max\left(\sum_n \|H_{l,n} W_l^\top\|_F^2, \epsilon_D\right)},$$

$$J_l(g) = S_l^+ D_{l,g}, \tag{11}$$

算法 1 任务敏感粒度分配

要求： f_θ， $\mathcal{D}_c$， $\mathcal{L}$， $\{64, 128\}$，以及 B_{scale}

确保： 分组分配 $\{g_l\}_{l \in \mathcal{L}}$

1: **对于** $n = 1$、.....、 N**执行**

2: 生成 y_n^* 并构建全部三个响应掩码

3: 对于任何空的词汇掩码，使用序列掩码

4: 教师强制 y_n^* 并缓存$H_{l,n}$ 和$U_{l,t,n}$

5: **结束循环**

6: 计算 $\widehat{E}_{l,t}$ 及其全局百分位排名

7: **对**每个 $l \in \mathcal{L}$**执行**

8: 计算S_l^+、$D_{l,64}$、$D_{l,128}$、 b_l，以及两个缩放因子成本

9: **结束循环**

10: 求解方程14 通过成本归一化动态规划

11: 当 $x_l = 1$且 $g_l = 128$时设置 $g_l = 64$，否则

12:**返回** $\{g_l\}_{l \in \mathcal{L}}$

$$C_l(g) = m_{\text{scale}} d_l^{\text{out}} \left\lceil \frac{d_l^{\text{in}}}{g} \right\rceil. \tag{12}$$

对于报告的 GUI-Q 布局，每个存储的缩放因子占用 $m_{\text{scale}} = 2$ 字节。对于 G64（$x_l = 1$），相较于 G128 的预测收益为 $b_l = [J_l(128) - J_l(64)]_+$。较高的收益需要实测失真降低和响应敏感度。TSGA 在不超出额外缩放因子存储的字节预算的前提下，使总收益最大化：

$$\max_{x_l \in \{0,1\}} \quad \sum_{l \in \mathcal{L}} b_l x_l, \tag{13}$$

$$\text{s.t.} \quad \sum_{l \in \mathcal{L}} [C_l(64) - C_l(128)] \, x_l \le B_{\text{scale}}. \tag{14}$$

当 G64 未降低代理指标时，正部算子将收益设为零。我们将整数成本除以它们的最大公约数，然后通过动态规划（DP）求解化简后的背包问题，详见算法 1。

3.3. 关系保持对称性选择

RPSS 根据目标选择所需的关系来选择网格。指令-视觉亲和度将截图特征与请求关联起来；空间加权视觉相似度则描述局部特征关系。保留这两者可补充响应敏感的分配。在 TSGA 确定 $\{g_l\}$ 后，RPSS 会评估一个冻结的 16 模块注意力池，该池涵盖浅层 q 投影以及三个深度上的 k/v/o 投影。注意力投影会直接影响这些关系，而冻结候选池则能限制性能分析开销。补充材料第 S2.8 节详细说明了候选池、平局处理顺序，以及视觉词元如何映射到屏幕截图。

对于候选项 $a = (b, p) \in \mathcal{C}$，RPSS 仅对$a$ 进行 g_a 量化，其余部分保留 BF16。这估算了网格在指定粒度下对关系保持的边际增益。它忽略了联合量化的投影之间的相互作用。保持相对亲和度无法保证下游动作不变，因为它们并未约束最终读出所使用的每个特征方向。我们在注意力和残差相加之后、在多层感知机之前的区块 b 处测量特征，

Ta表 1. UI-TARS-1.5-7B 准确率（%）。粗体/下划线：最佳/次佳 4位权重结果。BF16：全精度。

方法	SS-Pro TH	SS-v2 TH	AC TH	AC GA	AC SS	OSW-G TH	平均 TH
BF16	43.01	87.42	37.70	75.59	59.77	29.96	53.47
AWQ	22.77	66.19	<u>37.50</u>	65.82	<u>59.18</u>	4.43	31.13
GPTQ	18.98	69.89	25.00	58.20	46.09	1.42	30.09
MABA	0.51	33.96	15.43	49.02	44.34	2.84	12.44
VLM-PTQ	16.19	52.20	9.77	15.82	9.77	<u>6.03</u>	24.81
AutoRound	18.41	43.87	6.64	6.84	6.25	4.61	22.29
COD-TDQ	9.42	64.94	33.98	68.16	58.40	3.01	25.79
MBQ	17.14	51.18	8.79	14.84	8.79	4.96	24.43
QIG	15.64	50.24	7.81	13.87	7.81	4.08	23.32
Q-VLM	14.17	50.24	8.79	12.89	6.84	3.01	22.47
GuidedQuant	23.53	<u>71.93</u>	36.33	66.80	46.88	1.77	32.41
QVLA	15.62	65.72	35.94	71.09	51.17	4.43	28.59
ActQuant	<u>25.11</u>	71.86	36.52	<u>72.85</u>	55.66	2.66	<u>33.21</u>
GUI-Q	**35.62**	**85.91**	**37.70**	**74.41**	**59.57**	**15.74**	**45.76**

然后进行ℓ_2归一化。文本特征仅使用用户指令所在位置。视觉特征使用冻结投影器和原生令牌压缩之后可见的截图令牌。视觉令牌位置是其源图像块的归一化质心。设$T^r_{a,\ n} \in \mathbb{R}^{M_n \times d}$和$V^r_{a,\ n} \in \mathbb{R}^{V_n \times d}$分别为 $r \in \{\text{fp}, \text{sym}, \text{asym}\}$的文本和视觉特征。它们的归一化点积定义了每对令牌的亲和度：

$$A^r_{a,n,ij} = \frac{\langle T^r_{a,n,i}, V^r_{a,n,j} \rangle}{\tau}. \tag{15}$$

我们计算$P^r_{a,\ n} = \text{LogSinkhorn}(A^r_{a,\ n},\ \mu_n,\ \nu_n)$，其中$\mu_n$，$\nu_n \geq 0$、$\mathbf{1}^\top \mu_n = \mathbf{1}^\top \nu_n = 1$、$P^r_{a,\ n}\mathbf{1} = \mu_n$以及$(P^r_{a,\ n})^\top \mathbf{1} = \nu_n$。对数域归一化可避免不稳定的指数运算，并生成正的传输条目。我们使用这些耦合定义跨模态差异：

$$\mathcal{L}^{\text{cm}}_{a,n}(r) = \sum_{i=1}^{M_n} \sum_{j=1}^{V_n} P^{\text{fp}}_{a,n,ij} \log \frac{P^{\text{fp}}_{a,n,ij}}{P^r_{a,n,ij}}. \tag{16}$$

令$\rho_{n,i} \in [0,1]^2$为上文定义的视觉令牌位置。我们定义空间关系及其差异，然后使用二者的综合得分来选择非对称模块：

$$G^r_{a,n,ij} = \langle V^r_{a,n,i}, V^r_{a,n,j} \rangle \times \exp\left(-\frac{\|\rho_{n,i} - \rho_{n,j}\|_2^2}{2\sigma^2} \right), \tag{17}$$

$$\mathcal{L}^{\text{sp}}_{a,n}(r) = \frac{\|G^{\text{fp}}_{a,n} - G^r_{a,n}\|_F^2}{\|G^{\text{fp}}_{a,n}\|_F^2 + \epsilon_G}, \tag{18}$$

$$\mathcal{R}_{a,n}(r) = \mathcal{L}^{\text{cm}}_{a,n}(r) + \gamma \mathcal{L}^{\text{sp}}_{a,n}(r), \tag{19}$$

$$\Delta_a = \frac{1}{N} \sum_{n=1}^{N} \left[\mathcal{R}_{a,n}(\text{sym}) - \mathcal{R}_{a,n}(\text{asym}) \right], \tag{20}$$

$$\mathcal{A} = \arg \max_{\substack{\mathcal{S} \subseteq \mathcal{C} \\ |\mathcal{S}| \leq K_{\text{asym}}}} \sum_{a \in \mathcal{S}} [\Delta_a]_+. \tag{21}$$

RPSS 最多选择 K_{asym} 个得分为正的候选项；如果得分为正的候选项较少，则选择的模块也较少。若出现平分，则按固定的候选顺序处理。我们使用 $\tau = 0.1$、$\sigma = 0.25$、$\gamma = 1$ 和 $\epsilon_D = \epsilon_G = 10^{-8}$。这些边际分数需要根据我们所评估的图形用户界面任务中，最终联合量化模型的实际结果进行验证。

最终舍入校准。分配完成后，通用后端仅优化$[-1, 1]$中初始化为零的舍入偏移量。权重、以 BF16 存储的尺度、零点和常数组均保持不变。它使用所有有效的教师强制位置，在 50 个 Adam 步骤内最小化归一化的激活加权重建误差。计算采用 32 位浮点（FP32）算术和直通舍入。我们从 51 个迭代结果中，导出在整个校准缓存上达到最低误差的最早结果对应的硬编码值。

4. 实验

4.1. 实验设置

模型与评估指标。我们在 ScreenSpot-Pro（SS-Pro）、ScreenSpot-v2（SS-v2）、AndroidControl（AC）和 OSWorld-G（OSW-G）上评估 UI-TARS-1.5-7B、OS-Atlas-Base-7B 和 InternVL3.5-8B，且这些模型支持相应的输出格式。我们采用官方的 TH、定位准确率（GA）和步骤成功率（SS）指标。平均 TH 取三个共享基础定位基准的平均值；AndroidControl 的端点结果单独报告。闭环 AndroidWorld 使用 250 个冻结任务–重置对。严格成功要求奖励高于 0.5，且执行明确的终止动作。

训练后量化协议与基线方法。所有方法都将解码器的 q/k/v/o 和 gate/up/down 投影量化为 4 位权重，其余部分保留 BF16，并共用同一评估接口。我们比较 AWQ、GPTQ、AutoRound、VLM-PTQ、MABA、COD-TDQ、MBQ、QIG、Q-VLM，以及分配／动作感知迁移方法。

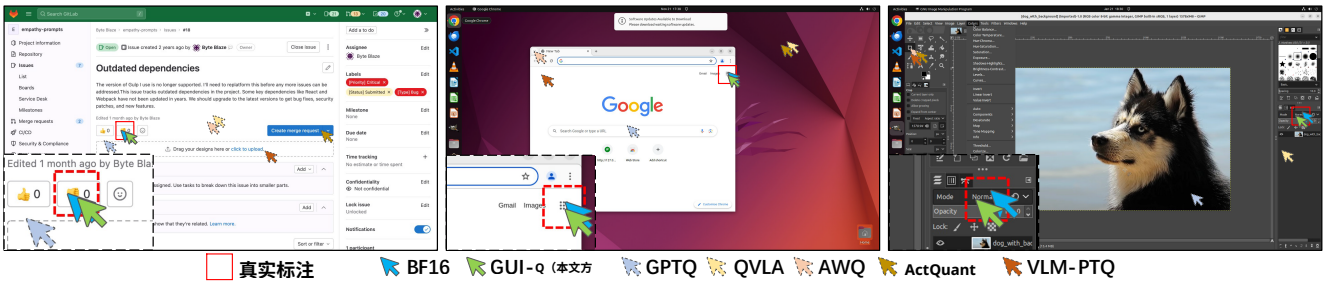


图 3. 图形用户界面任务上的定性比较。GUI-Q 在三个案例中均能正确定位所请求的控件。

各迁移方法共用校准数据、384 次反向传播、舍入后端和非对称上限。GUI-Q 使用 128 个无标签校准样例、256 个 token 的响应上限，以及 50 次舍入迭代。校准集、开发集、静态测试集和 AndroidWorld 彼此不重叠。词汇权重在单独标注的 512 个样例 OS-Atlas 开发集划分上选定，并在测试前冻结。只有校准过程不使用标签。静态 4 位权重结果是在固定测试样例上对五个配对随机种子取平均；BF16 结果则是确定性的。每项比较均固定提示词、图像处理、解析器、停止规则和官方评估器。受 GuidedQuant、QVLA 和 ActQuant 启发的迁移方法在此 4 位权重范围内实现分配或动作信号，而非复现所有上游方案。另设一个保留目标函数队列，以区分准则改进与适应效应。补充材料第 S2.2–S2.9 节详细说明了随机种子、基线设置、超参数、清单、数据划分和硬件。

配置和保留目标对照组使用单独的队列：12 个静态模型—工作负载对中的每一对包含 128 个案例，另有 250 个 AndroidWorld 案例，每个案例配有五个成对的种子槽位。在此，G_3 对 SS-Pro、SS-v2 和 OSW-G 上的 TH 取平均值（补充材料第 S12–S13 节）。在主要静态模型中，TSGA 分别为 56/196、20/196 和 81/252 个目标模块分配 G64；RPSS 分别选择 5、5 和 6 个非对称投影。其余目标模块则保留所分配的对称网格或 G128。补充材料第 S2.6 和 S6.2 节分别给出了各模型的字节上限和分组分配。

4.2. 基线比较

表 1显示，各基线方法在落地和动作保留方面表现不均。AWQ 和 GPTQ 的平均 TH 分别达到 31.13 和 30.09，但 OSW-G TH 仅为 4.43 和 1.42。VLM-PTQ 在所测试的 PTQ 基线中 OSW-G 表现领先，但 AndroidControl SS 仅为 9.77。GUI-Q 的平均 TH 达到 45.76，AndroidControl TH/GA/SS 分别为 37.70/74.41/59.57。下文的对照实验用于检验这些提升是否取决于评估标准。这些指标各有侧重：动作模式有效并不保证目标正确，而孤立点击准确也不能证明轨迹成功。我们报告 AndroidControl、闭环成功率和平均定位准确率。

在独立队列中，GUI-Q 在 UI-TARS 上分别比保留目标的 GuidedQuant 和 Action-Mixed-Fisher 对照组高 4.53

表 2. AgentCPM-GUI 在 250 个 AndroidWorld 回合上的表现。

方法	严格奖励		方法	严格奖励	
BF16	38.4%	96	AWQ	20.4%	54
GPTQ	17.6%	47	GuidedQuant	21.6%	57
ActQuant	23.2%	61	VLM-PTQ	15.6%	42
QVLA	24.0%	63	**GUI-Q**	**31.2%**	**81**

表3：在 UI-TARS 上进行的组件移除消融实验 .

配置	SS-Pro	SS-v2	OSW-G	AC GA	AC SS
GUI-Q	**35.62**	**85.91**	**15.74**	**74.41**	**59.57**
不使用 *TSGA*	11.57	65.57	13.83	55.86	40.43
不使用 *RPSS*	14.04	71.23	15.43	73.24	57.23

和 3.70 G_3 个百分点，在 AgentCPM-GUI 上的严格成功率分别高 5.36 和 4.08 个百分点。相应的 95% 案例聚类区间在 UI-TARS 上为 $[3.49, 5.57]$ 和 $[2.71, 4.69]$ 个百分点，在 Android-World 上为 $[3.84, 6.88]$ 和 $[2.64, 5.52]$。这些对比表明，性能提升并非仅来自常见的后端迁移对照。补充材料第 S12 节提供了该队列的所有终点和案例聚类区间。

闭环 AndroidWorld。 在相同的 250 组冻结任务–重置配对上，采用 GUI-Q 的 AgentCPM-GUI 达到 31.2% 的严格成功率，高于评估中表现最强的受 QVLA 启发的迁移方法（24.0%），但低于 BF16（38.4%）（表 2）。所有动作均可解析；“奖励”统计奖励成功。7.2 个百分点的提升反映了 GUI-Q 独有的 34 次成功与 QVLA 独有的 16 次成功（Holm 校正的精确 McNemar 检验 $p = 0.01535$）。未经校正的配对自助法 95% 区间为 $[1.6, 12.8]$ 个百分点；Bonferroni–Wald 同时 95% 区间为 $[-0.32, 14.72]$。补充材料第 S4.2 节说明了配对检验与区间方法之间的差异。

在相同任务上，五个随机种子带来7.2–7.6分的提升（均值：7.28）。一组不相交的250对复现得到30.8%对23.2%（+7.6 分），其中仅GUI-Q成功33次，仅QVLA成功14次（未经校正的精确 McNemar 检验$p = 0.00794$）。重复随机种子用于评估配置稳定性，并不会产生独立的任务样本。探索性错误表明坐标、邻近目标及动作字段失败更少，且错误分母各不相同。补充材料第S8节报告了独立复现结果、配对种子结果、任务分层和错误计数。

表4. UI-TARS上的准则替换消融实验。

分配目标	对称性准则	SS-Pro	SS-v2	OSW-G	平均 TH	AC GA	AC SS
GUI词汇代理指标	**GUI关系**	**35.62**	**85.91**	**15.74**	**45.76**	**74.41**	**59.57**
	注意力均方误差	33.19	84.10	12.94	43.41	73.24	58.59
通用终端损失	GUI关系	34.00	84.75	13.65	44.13	73.83	58.98
	注意力均方误差	28.61	74.43	10.60	37.88	65.00	51.88

4.3. 消融研究

表 3 与 4比较组件移除与准则替换，二者使用相同的校准数据、解码器目标、W4载荷、缩放因子字节上限、非对称上限、舍入后端和评估协议。阶段移除衡量其全部贡献，并可能改变实际元数据字节数。匹配字节数对照与准则替换在相同存储约束下评估选择质量。

为什么选择这些配置轴? 补充表S85（第S13节）在独立队列上检验了粒度和对称性。与G128-sym相比，G64-sym在UI-TARS、OS-Atlas和InternVL上的得分分别提高了 G_3 7.09、7.08和7.97个百分点。与固定G128相比，RPSS还分别增加了5.99、6.98和6.46 G_3 个百分点；在固定G64时，RPSS同样提高了这三个 G_3 得分。因此，更细粒度分组并不会消除对称性选择带来的收益。相较于均匀G64加RPSS，GUI-Q在使用更少解码器存储空间的同时，使得分变化G_3 为 $+0.42$、 $+0.26$和 -0.16 个百分点。这些彼此接近的点估计支持在该队列的元数据约束下进行选择性分配。但它们既不能证明准确率全面占优，也不能证明在所评估任务上具有统计等价性。

组件移除。 使用重新计算 RPSS 的均匀G128（无 *TSGA*）会使平均 TH 降低 15.44 个点，并使 AndroidControl GA/SS 降低 18.55/19.14 个点，同时移除任务感知分配及其额外尺度。保留 TSGA 分组但使所有模块保持对称（无 *RPSS*）会使平均 TH 降低 12.20 个点，动作变化则较小。与 RPSS 相关的最大降幅出现在 ScreenSpot-Pro 和 ScreenSpot-v2 上，这与其跨模态和局部空间标准一致。补充材料第S5.1节报告了每个移除阶段的端点对比和存储开销。

匹配字节数对照用于隔离选择质量。令 G_4对三个共享定位 TH 和 AndroidControl TH 取平均。选择 q_0–q_4 时，结果为 40.54，而 RPSS 的q_0、 q_1、 q_2、 q_3、 q_6为 43.74：配对增益为 3.21 个点（名义上的五个随机种子的 95% 区间：$[2.91, 3.51]$）。仅跨模态标准和仅空间标准分别达到 40.98 和 40.07；匹配字节数的随机选择达到 33.79。权重均方误差（MSE）和原始亲和度均方误差分别达到 35.58 和 40.39。这些比较支持组合关系标准。补充材料第S5.6节给出了随机选择所用全部五个随机种子的具体选择结果。

标准特异性与交互作用。 相较于采用注意力均方误差的通用终端损失，仅图形用户界面分配标准就增加了5.53个平均 TH 点数，仅关系标准增加了6.25个，而两者结合增加了7.88个（表4）。条件词汇/关系增益分别为1.63/2.35个点，经邦弗伦尼校正的五个随机种子95 % 区间为 $[0.40, 2.86]$/$[1.28, 3.41]$。它们的 -3.90点交互作用（$[-5.64, -2.17]$）表明收益存在重叠；两个条件增益仍均为正值。这些区间量化了固定测试集上的随机种子变化。补充材料第S5.2节对主效应与交互作用进行了完整分析。

排序诊断。在采用 RPSS 之前，匹配字节数的全对称对照组中，TSGA 的得分为 $G_4 = 33.67$，而随机分配和单独使用失真时分别为 29.82 和 30.71。TSGA 在 29.8 分钟内达到 0.57 的预言机排序相关性，并获得预言机收益的 84%。精确序列 Fisher、Hutchinson 曲率和预言机搜索分别在 42.5、83.4 和 173.6 分钟内达到 34.02、34.10 和 34.46。这些对照组体现了因式分解得分在性能分析与准确率之间的权衡。纳入前缀位置后，平均 TH 从 45.76 变为 45.86，存储量则保持不变。RPSS 的预言机排序相关性为 0.72，重叠率为 82%，正向精确率为 84%。补充材料第 S5.3 节和第 S5.6 节报告了对从校准数据中抽取的随机种子和重复样本所进行的排序与计算诊断。

联合 W4 验证。 已测得的 UI-TARS 部署五投影集合成对修正总计 -1.39 G_4 分（占单项增益总和的 12.07%）。二阶预测为 43.83，实测为 43.74，但对固定 early-q 性能高估了 3.22 分。直接联合评估仍然必不可少。在 21 组等字节搜索中，有一次交换比 RPSS 高 0.20 分，名义上的五种子 95% 区间为 $[-0.20, 0.61]$；这项微小提升在统计学上尚无法判定。受限的 OS-Atlas 和 InternVL 搜索使平均 TH 分别提高至多 0.14 和 0.12 分。补充材料第 S10 节列出了全部 16 个单项、120 个实测配对项和跨模型搜索。在 UI-TARS、OS-Atlas 和 InternVL 上，符合条件的投影分别为 112、112 和 144 个；冻结候选池覆盖了完整搜索中排名前 K 项的 4/5、4/5 和 5/6。完整搜索在这些检查中仅使报告的聚合指标提高 0.13/0.13/0.12 分，支持了实用候选池的有效性，同时也表明其覆盖率取决于任务。在 UI-TARS 的尺度字节上限 12.80/21.76 MB 下，精确联合枚举分别测试了 1,368/2,394 种配置。其表现比顺序选择高 0.18/0.14 G_4 分。补充材料第 S6.4 节详细说明了各预算下的搜索空间。

词汇掩码与校准鲁棒性。 在 UI-TARS 上，加权词汇代理指标的平均 TH 为 45.76，而均匀权重为 45.08，使用坐标解析器为 45.59，使用动作模式解析器为 45.93，且预算上限相同。在计算量匹配时，序列、仅数值和仅非数值目标分别为 44.08、44.69 和 43.72；匹配的随机掩码为 40.04。经审计的数字掩码精确率/召回率为 73.29/88.52%，回退率为 5.47 %。权重扰动 ± 0.05 后，平均 TH 为 45.45–45.63。固定权重与均匀权重在 OS-Atlas 上分别为 35.36/34.69，在 InternVL 上分别为 49.45/48.95，支持权重的迁移能力。在 AgentCPM-GUI 上，这项比较得到 78/250 对 74/250 的严格成功，其中加权方案独有 9 次成功，均匀权重方案独有 5 次成功（精确 McNemar 检验 $p = 0.42395$）。这些权重在所测设置中保持稳定，但其相较于均匀权重的闭环优势在统计学上仍无法判定。补充材料第 S5.4–S5.5 节和第 S9 节报告了词元审计及词汇权重测试。

使用 64、128 和 256 个校准样例，分别得到 44.92、45.76 和 45.86 的平均 TH。性能分析耗时分别为 27.97、54.43 和 107.61 分钟。在 $N = 128$ 时，单域校准的结果为 44.96–45.41，而混合域为 45.76。这些结果表明，使用 128 个混合域样例是兼顾准确率与成本的折衷方案。补充材料第 S6.1 节报告了所有校准样例数量及源域。

RPSS 运行设置。分别单独调整温度、关系权重和半径进行参数扫描，结果为 $G_4 = 43.09$– 43.74。使用 10、20 和 30 次 Sinkhorn 迭代时，G_4 分别为 43.54、43.74 和 43.73。将非对称模块数量上限设为 3、5 和 7 时，结果分别为 42.91、43.74 和 43.63。超过 20 次迭代或使用五个非对称投影，并不能提升报告的 UI-TARS 得分。补充材料第 S6.3 节列出了完整的参数扫描结果和性能分析成本。

4.4. 定性与可视化分析

图 3展示了相邻目标、小物体和语义干扰项导致的错误。GPTQ接近GitLab的反应对照组，但漏掉了踩按钮。小型的Google应用网格图标会与Chrome菜单和桌面杂乱元素相互干扰。在GIMP中，展开的菜单分散了对狭窄图层模式下拉菜单的注意力。GUI-Q选中了预期的全部三个对照组。这些示例说明了研究动机；受控消融实验提供了定量证据。补充图S1和第S3.3节给出了更多图形用户界面目标示例。

4.5. 跨模型泛化

在三个共享基础定位基准上，UI-TARS 在 BF16 下的平均 TH 为 53.47，采用受 ActQuant 启发的迁移方法时为 33.21，采用 GUI-Q 时为 45.76。OS-Atlas 在相同设置下的得分分别为 39.22、25.74 和 35.36。InternVL 在 BF16 下得分为 50.21，采用其表现最强的基线方法 AWQ 时为 41.41，采用 GUI-Q 时为 49.45。相较于 4位权重基线方法，得分分别提升了 12.55、9.62 和 8.04 个百分点，与 BF16 的差距分别为 7.71、3.86 和 0.76 个百分点。GUI-Q 提升了全部三个检查点的表现，但全精度准确率的恢复程度各不相同。

GUI-Q在OS-Atlas上的SS-Pro/SS-v2/OSW-G TH为11.76/74.03/20.28，在InternVL上为22.45/87.74/38.16。在OS-Atlas上，它在每项基准测试中都超过了测试过的最佳训练后量化基线。InternVLAndroidControlSS为69.77，而BF16和AWQ分别为62.89和56.99。补充材料第S3节提供了各数据集的比率和命中总数。动态分辨率检查保留了这些增益（补充材料第S2.8节）：在UI-TARS上，GUI-Q和所列对比方法的平均TH依次为51.72/41.02；在OS-Atlas上依次为37.93/29.88。

在单独保留目标函数的队列中，GUI-Q 在 OS-Atlas 上比 Action-Mixed-Fisher 高出 3.28 G_3 个百分点，在 InternVL 上高出 4.06 个百分点。相应的 95% 案例聚类区间分别为 $[2.40, 4.22]$ 和 $[3.13, 5.05]$（补充材料第 S12 节）。这项比较检验了在保留基线目标函数时，性能提升是否依然存在。这些提升是在该队列中测得的，因此应与主要静态测试中的提升分开解读。

4.6. 效率分析

我们报告了一次性配置成本、静态峰值分配量以及打包 W4模型调用延迟。使用 128 个校准样本时，UI-TARS 的 TSGA 配置耗时 29.80 分钟，RPSS 耗时 24.63 分钟，峰值为 15.43 吉比字节；DP 耗时 0.138 秒。静态导出文件中，UI-TARS、OS-Atlas 和 InternVL 的每个目标权重分别占用 4.166、4.145 和 4.172 位，其中包含量化器元数据。补充材料第S7节列出了缩放因子、零点、描述符、BF16剩余部分及序列化字节数。同一节还对 AgentCPM-GUI 导出文件进行了单独审计，并与静态导出文件一并列出。

GUI-Q 将 UI-TARS 的峰值内存分配从 16.62 吉比字节降至 8.32 吉比字节，将 OS-Atlas 从 15.75 吉比字节降至 6.93 吉比字节，并将 InternVL 从 16.67 吉比字节降至 7.29 吉比字节：降幅分别为 49.9%、56.0% 和 56.2%。从视觉编码到解码的固定 BF16 输出回放分别实现了 1.92–2.80×、1.70–2.80× 和 1.70–2.24× 倍的加速。在 AndroidWorld 的 250 个输入上进行首次决策回放时，耗时从 792.04 毫秒降至 597.08 毫秒（1.33× 倍），峰值内存减少了 56.67%。这些 4位权重与 BF16 之间的收益涉及打包模型调用；在 4位权重配置内部，对照组显示出的计时差异要小得多。固定 BF16 参考输出长度，可以将环境执行速度与动作变化及提前停止区分开来。逻辑分派覆盖率和降维数值检查验证了在经过测试的算子形状和数值容差下的执行。补充章节 S7.6 和 S14 报告了工作负载计时、配对自助法区间、分派覆盖率以及执行路径的数值检查。AndroidWorld 回放加速的配对自助法 95% 区间为 $[1.29, 1.36]\times$；2.80× UI-TARS OSWorld-G 加速的 95% 区间为 $[2.67, 2.92]\times$。这些区间支持在固定输出长度下实现更快的模型调用。不包含环境执行。因此，测得的加速反映的是模型调用，并不意味着代理在真实环境中完成一项图形用户界面任务所需的时间，

而且也不描述代理在真实环境中完成一项图形用户界面任务需要多长时间。

5. 结论

GUI-Q 将 TSGA 和 RPSS 结合起来，用于仅解码器 W4A16 量化。TSGA 根据响应敏感度和失真降低分配更细粒度的分组，在尺度字节预算内保留图形用户界面响应。RPSS 选择对称或非对称网格，以保留目标选择所需的指令–视觉关联和局部空间关系。校准使用冻结模型自身的响应，无需任务标签。与评估的 W4 基线方法相比，GUI-Q 提升了落地效果，并在单独的闭环评估中提高了任务成功率。与 BF16 相比，静态峰值 GPU 内存降低了 49.9–56.2%。在重放相同固定输出时，打包模型调用的运行速度提升了 1.7–2.8× 倍。

致谢

本研究部分获得天津市科技重大专项（编号：25ZXRGGX00120）、国家自然科学基金委员会（编号：62576176）以及中央高校基本科研业务费专项资金（南开大学，编号：070-63253235）的资助。计算资源由南开大学超级计算中心（NKSC）提供支持。

大语言模型使用声明

ChatGPT（OpenAI）用于协助润色论文正文和补充材料中的语言，并调整 LaTeX 格式。核心研究理念和方法设计均由作者独立完成，未借助大语言模型。作者审阅了所有经 AI 辅助生成的内容，并对稿件承担全部责任。

参考文献

[1] Arash Akbari、Arman Akbari、Masih Eskandar、Qitao Tan、Yixiao Chen、Jingwu Luo、Bertha Pangaribuan、Liyun Zhang、Jennifer Dy、Geng Yuan 等。ActQuant：面向视觉-语言-动作模型的亚 4 位动作引导量化。*arXiv* 预印本 *arXiv:2605.24011*，2026年。[2] Gilles Baechler、Srinivas Sunkara、Maria Wang、Fedir Zubach、Hassan Mansoor、Vincent Etter、Victor Carbune、Jason Lin、Jindong Chen 和 Abhanshu Sharma。Screenai：用于用户界面和信息图理解的视觉-语言模型。载于第三十三届国际人工智能联合会议论文集，第 3058–3068 页，2024年。[3] Chongyang Bai、Xiaoxue Zang、Ying Xu、Srinivas Sunkara、Abhinav Rastogi、Jindong Chen 和 Blaise Agüera y Arcas。Uibert：学习用于用户界面理解的通用多模态表征。载于第三十届国际人工智能联合会议论文集，第 1705–1712 页，2021年。

[4] Shubhang Bhatnagar、Andy Xu、Kar-Han Tan 和 Narendra Ahuja。Luq：面向多模态大型语言模型的逐层超低比特量化。《机器学习研究汇刊》，2026年。[5] Kanzhi Cheng、Qiushi Sun、Yougang Chu、Fangzhi Xu、Yantao Li、Jianbing Zhang 和 Zhiyong Wu。Seeclick：利用 GUI 定位实现高级视觉图形用户界面智能体。在《第*62*届计算语言学协会年会论文集》（第*1*卷：长篇论文），第9313–9332页，2024年。[6] Wenhua Cheng、Weiwei Zhang、Haihao Shen、Yiyang Cai、Xin He、Lv Kaokao 和 Yi Liu。通过符号梯度下降优化权重舍入，实现大语言模型量化。在计算语言学协会论文集：*EMNLP 2024* 论文集，第11332–11350页，2024年。[7] Wenhua Cheng、Weiwei Zhang、Heng Guo、Haihao Shen 和 Zaner Ma。SignRoundV2：弥合大语言模型极低比特训练后量化中的性能差距。*arXiv* 预印本 *arXiv:2512.04746*，2026年。2026年5月修订，第2版。[8] Juncan Deng 和 Kejie Huang。VLM-PTQ：大型视觉语言模型的高效训练后量化。在 *IEEE/CVF* 计算机视觉与模式识别会议论文集，第24696–24705页，2026年。[9] Xiang Deng、Yu Gu、Boyuan Zheng、Shijie Chen、Sam Stevens、Boshi Wang、Huan Sun 和 Yu Su。Mind2web：迈向面向网页的通用型智能体。神经信息处理系统进展，36：28091–28114，2023年。[10] Zhen Dong、Zhewei Yao、Daiyaan Arfeen、Amir Gholami、Michael W Mahoney 和 Kurt Keutzer。Hawq-v2：神经网络的海森矩阵感知迹加权量化。神经信息处理系统进展，33：18518–18529，2020年。[11] Ali Edalati、Alireza Ghaffari、Mahsa Ghazvini Nejad、Lu Hou、Boxing Chen、Masoud Asgharian 和 Vahid Partovi Nia。Oac：面向精确训练后量化的输出自适应校准。在*AAAI*人工智能会议论文集，第16453–16461页，2025年。[12] Elias Frantar、Saleh Ashkboos、Torsten Hoefler 和 Dan Alistarh。Optq：面向生成式预训练变换器的精确训练后量化。在第十一届国际学习表征会议，2023年。会议标题为 OPTQ，也通常称为 GPTQ。[13] Jinuk Kim、Marwa El Halabi、Wonpyo Park、Clemens JS Schaefer、Deokjae Lee、Yeonhong Park、Jae W Lee 和 Hyun Oh Song。GuidedQuant：利用终端损失引导进行大语言模型量化。在第*42*届国际机器学习会议，第30011–30037页。PMLR，2025年。[14] Jing Yu Koh、Robert Lo、Lawrence Jang、Vikram Duvvur、Ming Lim、Po-Yu Huang、Graham Neubig、Shuyan Zhou、Russ Salakhutdinov 和 Daniel Fried。Visualwebarena：评估真实视觉网页任务中的多模态智能体。在《第*62*届计算语言学协会年会论文集》（第*1*卷：长篇论文），第881–905页，2024年。[15]Kaixin Li、Ziyang Meng、Hongzhan Lin、Ziyang Luo、Yuchen Tian、Jing Ma、Zhiyong Huang 和 Tat-Seng

Chua。Screenspot-pro：面向专业高分辨率计算机使用的 GUI 定位。在第*33*届 *ACM* 国际多媒体会议论文集，第8778–8786页，2025年。[16] Shiyao Li、Yingchun Hu、Xuefei Ning、Xihui Liu、Ke Hong、Xiaotao Jia、Xiuhong Li、Yaqi Yan、Pei Ran、Guohao Dai 等。Mbq：面向大型视觉语言模型的模态平衡量化。在*IEEE/CVF* 计算机视觉与模式识别会议论文集（*CVPR*），第4167–4177页，2025年。[17] Tianqi Li、Wenyu Fang、Xin He、Xue Geng、Xu Cheng 和 Yun Liu。当 W4A4 遇上伪装目标检测：令牌组双约束激活量化。在计算机视觉——*ECCV 2026*，第*XIX*部分，第77–95页。Springer，2026年。[18] Wei Li、William Bishop、Alice Li、Chris Rawles、Folawiyo Campbell-Ajala、Divya Tyamagundlu 和 Oriana Riva。论数据规模对 UI 控制代理的影响。神经信息处理系统进展，37：92154，2024年。[19] Ji Lin、Jiaming Tang、Haotian Tang、Shang Yang、Wei-Ming Chen、Wei-Chen Wang、Guangxuan Xiao、Xingyu Dang、Chuang Gan 和 Song Han。AWQ：面向端侧大语言模型压缩与加速的激活感知权重量化。在机器学习与系统论文集，2024年。[20] Markus Nagel、Rana Ali Amjad、Mart Van Baalen、Christos Louizos 和 Tijmen Blankevoort。向上还是向下？面向训练后量化的自适应舍入。在国际机器学习会议，第7197–7206页。PMLR，2020年。[21] Yujia Qin、Yining Ye、Junjie Fang、Haoming Wang、Shihao Liang、Shizuo Tian、Junda Zhang、Jiahao Li、Yunxin Li、Shijue Huang 等。UI-TARS：利用原生代理开创自动化 GUI 交互。*arXiv* 预印本 *arXiv:2501 12326*，2025年。[22] Chris Rawles、Sarah Clinckemaillie、Yifan Chang、Jonathan Waltz、Gabrielle Lau、Marybeth Fair、Alice Li、William Bishop、Wei Li、Folawiyo Campbell-Ajala 等。Androidworld：面向自主智能体的动态基准测试环境。在国际学习表征会议，2025年。[23] Wenqi Shao、Mengzhao Chen、Zhaoyang Zhang、Peng Xu、Lirui Zhao、Zhiqian Li、Kaipeng Zhang、Gao Peng、Yu Qiao 和 Ping Luo。OmniQuant：面向大型语言模型的全向校准量化。在国际学习表征会议，2024年。[24] Changyuan Wang、Ziwei Wang、Xiuwei Xu、Yansong Tang、Jie Zhou 和 Jiwen Lu。Q-VLM：大型视觉语言模型的训练后量化。神经信息处理系统进展，37：114553–114573，2024年。[25] Weiyun Wang、Zhangwei Gao、Lixin Gu、Hengjun Pu、Long Cui、Xingguang Wei、Zhaoyang Liu、Linglin Jing、Shenglong Ye、Jie Shao 等。InternVL3.5：提升开源多模态模型的通用性、推理能力与效率。*arXiv* 预印本 *arXiv:2508.18265*，2025年。[26] Zhiyong Wu、Zhenyu Wu、Fangzhi Xu、Yian Wang、Qiushi Sun、Chengyou Jia、Kanzhi Cheng、Zichen Ding、Liheng Chen、Paul Pu Liang 等。OS-Atlas：基础动作通用型 GUI 智能体的模型。在国际学习表征会议，2025年。[27] Ziwei Xiang、Fanhu Zeng、Hongjian Fang、Rui-Qi Wang、Renxing Chen、Yanan Zhu、Yi Chen、Peipei Yang 和 Xu- Yao Zhang。利用量化感知积分梯度对大型视觉语言模型进行细粒度训练后量化。在*IEEE/CVF*计算机视觉与模式识别会议（*CVPR*）论文集，第 3500–3510 页，2026年。[28] Tianbao Xie、Danyang Zhang、Jixuan Chen、Xiaochuan Li、Siheng Zhao、Ruisheng Cao、Toh J Hua、Zhoujun Cheng、Dongchan Shin、Fangyu Lei 等。OSWorld：在真实计算机环境中为开放式任务对多模态智能体进行基准测试。神经信息处理系统进展，37：52040–52094，2024年。[29] Tianbao Xie、Jiaqi Deng、Xiaochuan Li、Junlin Yang、Haoyuan Wu、Jixuan Chen、Wenjing Hu、Xinyuan Wang、Yuhui Xu、Zekun Wang 等。通过用户界面分解与合成扩展计算机使用基础。在神经信息处理系统进展，2025年。数据集与基准赛道。[30] Yuhao Xu、Yantai Yang、Zhenyang Fan、Yufan Liu、Yuming Li、Bing Li 和 Zhipeng Zhang。QVLA：视觉-语言-动作模型量化中的通道并非都同等重要。在第十四届国际学习表征会议，2026年。[31] Tom Yeh、Tsung-Hsiang Chang 和 Robert C Miller。Sikuli：使用 GUI 截图进行搜索和自动化。在第 *22* 届 *ACM* 用户界面软件与技术年度研讨会论文集，第 183–192 页，2009年。[32] JiangYong Yu、Sifan Zhou、Dawei Yang、Shuoyu Li、Shuo Wang、Xing Hu、Chen Xu、Zukang Xu、Changyong Shu 和 Zhihang Yuan。MQuant：通过静态量化释放多模态大语言模型的推理潜力。在第 *33* 届 *ACM* 国际多媒体会议论文集，第 1783–1792 页，2025年。[33] Jingxuan Zhang、Yunta Hsieh、Zhongwei Wan、Haokun Lin、Xin Wang、Ziqi Wang、Yingtie Lei 和 Mi Zhang。QuantVLA：面向视觉-语言-动作模型的尺度校准训练后量化。在*IEEE/CVF*计算机视觉与模式识别会议论文集，第 39539–39549 页，2026年。[34] Xi Zhang、Hanwei Zhu、Jiamang Wang、Xiaolin Wu 和 Weisi Lin。面向视觉语言模型混合精度量化的模态感知比特分配。在*IEEE/CVF*计算机视觉与模式识别会议论文集（*CVPR*）论文发现，第 9305–9315 页，2026年。[35] Zhong Zhang、Yaxi Lu、Yikun Fu、Yupeng Huo、Shenzhi Yang、Yesai Wu、Han Si、Xin Cong、Haotian Chen、Yankai Lin 等。AgentCPM-GUI：通过强化微调构建移动端使用智能体。在*2025*年自然语言处理经验方法会议论文集：系统演示，第 155–180 页，2025年。[36] Shuyan Zhou、Frank F Xu、Hao Zhu、Xuhui Zhou、Robert Lo、Abishek Sridhar、Xianyi Cheng、Tianyue Ou、Yonatan Bisk、Daniel Fried 等。WebArena：用于构建自主代理的真实网络环境。在国际学习表征会议，2024年。